High order discretization of seismic waves-problems based upon DG-SE methods[†]

Hélène Barucq[1], Henri Calandra[2], Aurélien Citrain[3,1], Julien Diaz[1] and Christian Gout[3]

[1] Team project Magique.3D, INRIA, E2S UPPA, CNRS, Pau, France.

[2] TOTAL SA, CSTJF, Pau, France.

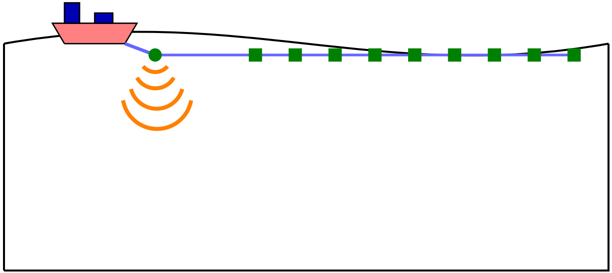
[3] LMI, Normandie Université, INSA ROUEN, 76000 Rouen, France.

WAVES 2019

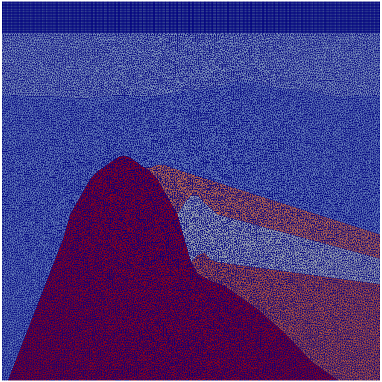
The authors thank the M2NUM project which is co-financed by the European Union with the European regional development fund (ERDF, HN0002137) and by the Normandie Regional Council.

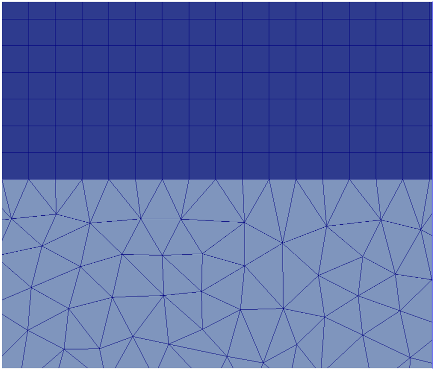
[†] *This work is dedicated to the memory of Dimitri Komatitsch.*

Seismic imaging



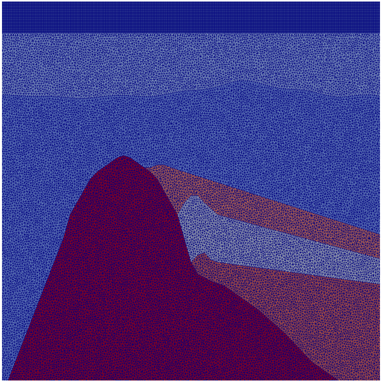
Why using hybrid meshes?

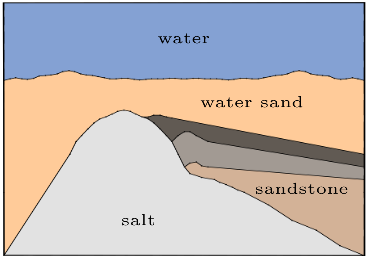




- Useful when the use of unstructured grid is non-sense (e.g. medium with a layer of water).
- Well suited for the coupling of numerical methods in order to reduce the computational cost and improve the accuracy.

Why using hybrid meshes?





- Useful when the use of unstructured grid is non-sense (e.g. medium with a layer of water).
- Well suited for the coupling of numerical methods in order to reduce the computational cost and improve the accuracy.

Elastodynamic system

$x \in \Omega \subset \mathbb{R}^d$, $t \in [0, T]$, $T > 0$:

$$\begin{cases} \rho(x)\dfrac{\partial v}{\partial t}(x,t) & = \nabla \cdot \underline{\underline{\sigma}}(x,t), \\ \dfrac{\partial \underline{\underline{\sigma}}}{\partial t}(x,t) & = \underline{\underline{C}}(x)\underline{\underline{\epsilon}}(v(x,t)). \end{cases}$$

With:

- $\rho(x)$ the density,
- $\underline{\underline{C}}(x)$ the elasticity tensor,
- $\underline{\underline{\epsilon}}(x,t)$ the deformation tensor,
- $v(x,t)$, the wavespeed,
- $\underline{\underline{\sigma}}(x,t)$ the strain tensor.

Elasticus software

Software written in **Fortran** for wave propagation simulation in the **time domain**

Features

Simulation:

- on various types of meshes (**unstructured triangles and tetrahedra**),
- on **heterogeneous** media (**acoustic, elastic and elasto-acoustic**).

- **Discontinuous Galerkin** (DG) based on **unstructured triangles and unstructured tetrahedra**,
- with various time-schemes : **Runge-Kutta (2 or 4), Leap-Frog**,
- with **multi-order** computation(**p-adaptivity**)...

Table of contents

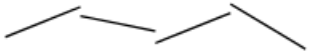
1. DGm and SEm
2. Comparison DG/SEM on structured quadrangle mesh
3. DG/SEM coupling
4. DGSEM vs DG
5. 3D extension
6. Perfectly Matched Layer(PML)

1. DGm and SEm
   - Discontinous Galerkin Method (DG)
   - Spectral Element Method (SEM)
   - Advantages of each method

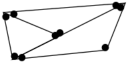
Use discontinuous functions :


mesh

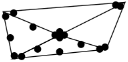

continuous


discontinuous

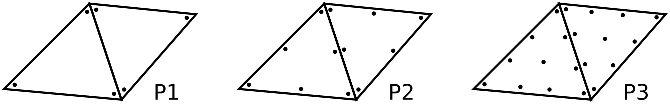
h adaptivity :



p adaptivity :

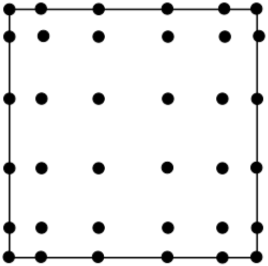


Degrees of freedom necessary on each cell :



General principle

- Finite Element Method (FEM) discretization + Gauss-Lobatto quadrature,
- Gauss-Lobatto points as degrees of freedom (gives us exponential convergence on L^2-norm).



- $\displaystyle\int f(x)dx \approx \sum_{j=1}^{N+1} \omega_j f(\xi_j)$,
- $\varphi_i(\xi_j) = \delta_{ij}$.

Advantages of each method

DG

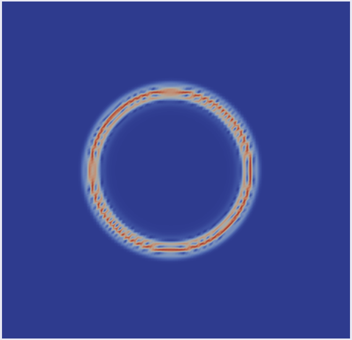
- Element per element computation (*hp*-adaptivity).
- Time discretization quasi explicit (block diagonal mass matrix).
- Simple to parallelize.
- Robust to brutal changes of physics and geometry

SEM

- Couples the flexibility of FEM with the accuracy of the pseudo-spectral method.
- Simplifies the mass and stiffness matrices (mass matrix diagonal).

2. Comparison DG/SEM on structured quadrangle mesh
   - Description of the test cases
   - Comparative tables

Physical parameters



P wavespeed	$1000\ m.s^{-1}$
Density	$1\ kg.m^{-3}$

Second order **Ricker Source** in *Pwave* ($f_{peak} = 10Hz$)

General context

- **Acoustic homogeneous** medium.
- Four different meshes : **10000 cells, 22500 cells**, 90000 cells, 250000 cells.
- CFL computed using **power iteration** method.
- **Leap-Frog** time scheme.
- **Eight threads** parallel execution with **OpenMP**.

Comparative tables

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time	Nb of time steps
DG($\alpha = 0$)	3.18e-3	2e-1	5.13	629
SEM	4.9e-3	5e-2	0.80	409

Table: DG not penalized and SEM comparison on the 10000 cells case

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0$)	2.12e-3	7e-1	18.11	943
SEM	3.26e-3	4e-2	3.54	613

Table: DG not penalized and SEM comparison on the 20000 cells case

Comparative tables

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time	Nb of time steps
DG($\alpha = 0$)	3.18e-3	2e-1	5.13	629
SEM	4.9e-3	5e-2	0.80	409

Table: DG not penalized and SEM comparison on the 10000 cells case

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0$)	2.12e-3	7e-1	18.11	943
SEM	3.26e-3	4e-2	3.54	613

Table: DG not penalized and SEM comparison on the 20000 cells case

Comparative tables

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time	Nb of time steps
DG($\alpha = 0$)	3.18e-3	2e-1	5.13	629
SEM	4.9e-3	5e-2	0.80	409

Table: DG not penalized and SEM comparison on the 10000 cells case

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0$)	2.12e-3	7e-1	18.11	943
SEM	3.26e-3	4e-2	3.54	613

Table: DG not penalized and SEM comparison on the 20000 cells case

Comparative tables

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time	Nb of time steps
DG($\alpha = 0.5$)	2e-3	3e-2	7.93	1000
SEM	4.9e-3	5e-2	0.80	409

Table: DG penalized and SEM comparison on the 10000 cells case

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0.5$)	1.33e-3	2e-2	32.98	1502
SEM	3.26e-3	4e-2	3.54	613

Table: DG penalized and SEM comparison on the 20000 cells case

Comparative tables

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time	Nb of time steps
DG($\alpha = 0.5$)	2e-3	3e-2	7.93	1000
SEM	4.9e-3	5e-2	0.80	409

Table: DG penalized and SEM comparison on the 10000 cells case

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0.5$)	1.33e-3	2e-2	32.98	1502
SEM	3.26e-3	4e-2	3.54	613

Table: DG penalized and SEM comparison on the 20000 cells case

Comparative tables

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time	Nb of time steps
DG($\alpha = 0.5$)	2e-3	3e-2	7.93	1000
SEM	4.9e-3	5e-2	0.80	409

Table: DG penalized and SEM comparison on the 10000 cells case

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0.5$)	1.33e-3	2e-2	32.98	1502
SEM	3.26e-3	4e-2	3.54	613

Table: DG penalized and SEM comparison on the 20000 cells case

Comparative tables

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time	Nb of time steps
DG($\alpha = 0.5$)	2e-3	3e-2	7.93	1000
SEM	4.9e-3	5e-2	0.80	409

Table: DG penalized and SEM comparison on the 10000 cells case

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0.5$)	1.33e-3	2e-2	32.98	1502
SEM	3.26e-3	4e-2	3.54	613

Table: DG penalized and SEM comparison on the 20000 cells case

Comparative tables

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0.5$)	2e-3	3e-2	7.93	1000
SEM	2e-3	3e-2	2.12	1000

Table: DG penalized and SEM comparison using the same CFL on a 10000 thousands cells mesh

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG ($\alpha = 0.5$)	1.33e-3	2e-2	32.98	1502
SEM	1.33e-3	2e-2	8.67	1502

Table: DG penalized and SEM comparison using the same CFL on a 20000 thousands cells mesh

Comparative tables

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0.5$)	2e-3	3e-2	7.93	1000
SEM	2e-3	3e-2	2.12	1000

Table: DG penalized and SEM comparison using the same CFL on a 10000 thousands cells mesh

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG ($\alpha = 0.5$)	1.33e-3	2e-2	32.98	1502
SEM	1.33e-3	2e-2	8.67	1502

Table: DG penalized and SEM comparison using the same CFL on a 20000 thousands cells mesh

Comparative tables

- Error computed as the difference between an analytical and a numerical solution for each method.
- Three cases considered : DG without penalization terms, DG with penalization terms and SEM.

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG($\alpha = 0.5$)	2e-3	3e-2	7.93	1000
SEM	2e-3	3e-2	2.12	1000

Table: DG penalized and SEM comparison using the same CFL on a 10000 thousands cells mesh

	CFL	L2-error	CPU-time(s)	Nb of time steps
DG ($\alpha = 0.5$)	1.33e-3	2e-2	32.98	1502
SEM	1.33e-3	2e-2	8.67	1502

Table: DG penalized and SEM comparison using the same CFL on a 20000 thousands cells mesh

Advantages of each method

DG

- Element per element computation (*hp*-adaptivity).
- Time discretization quasi explicit (block diagonal mass matrix).
- Simple to parallelize.
- Robust to brutal changes of physics and geometry

SEM

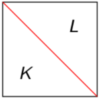
- Couples the flexibility of FEM with the accuracy of the pseudo-spectral method.
- Simplifies the mass and stiffness matrices (mass matrix diagonal).
- **Reduces the computational costs on structured quadrangle cells in comparison with DG**

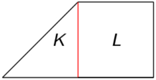
3. DG/SEM coupling
   - Hybrid meshes structures
   - Variational formulation

Hybrid meshes structures

- Aim at coupling P_k and Q_k structures.
- Need to extend or split some structures (e.g. neighbour indices).
- Define new face matrices:

$$M_{ij}^{K,L} = \int_{K\cap L} \phi_i^K \phi_j^L, \quad M_{ij}^{K,L} = \int_{K\cap L} \psi_i^K \psi_j^L, \quad M_{ij}^{K,L} = \int_{K\cap L} \phi_i^K \psi_j^L.$$

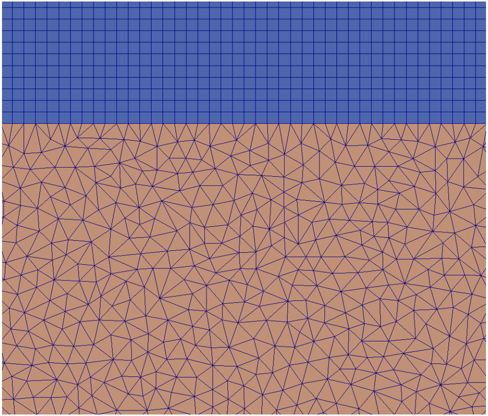




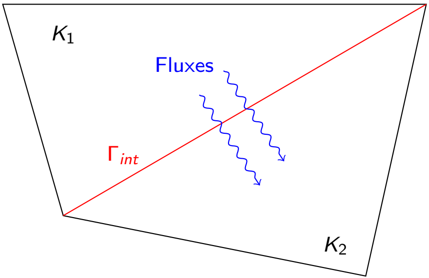
Variational formulation

Global context

- Domain in two parts : $\Omega_{h,1}$ (**structured quadrangles + SEM**), $\Omega_{h,2}$ (**unstructured triangles + DG**).



Variational formulation



Definitions

- Jump and average.

$$[[u]] = (u_{K_1}\mathbf{n}_{K_1} + u_{K_2}\mathbf{n}_{K_2})$$

$$\{\{u\}\} = \frac{1}{2}(u_{K_2} + u_{K_1})$$

SEM variational formulation :

$$\begin{cases} \displaystyle\int_{\Omega_{h,1}} \rho \partial_t v_1 \cdot w_1 = -\int_{\Omega_{h,1}} \sigma_1 : \nabla w_1 + \int_{\Gamma_{out,1}} (\sigma_1 \mathbf{n}_1) \cdot w_1, \\ \displaystyle\int_{\Omega_{h,1}} \partial_t \sigma_1 : \xi_1 = \int_{\Omega_{h,1}} (C\xi_1) : \nabla v_1. \end{cases}$$

DG variational formulation :

$$\begin{cases} \displaystyle\int_{\Omega_{h,2}} \rho \partial_t v_2 \cdot w_2 = -\int_{\Omega_{h,2}} \sigma_2 : \nabla w_2 + \int_{\Gamma_{out,2}} (\sigma_2 \mathbf{n}_2) \cdot w_2 + \int_{\Gamma_{int}} \{\{\sigma_2\}\} : [[w_2]], \\ \displaystyle\int_{\Omega_{h,2}} \partial_t \sigma_2 : \xi_2 = -\int_{\Omega_{h,2}} (\boldsymbol{\nabla} \cdot (C\xi_2)) \cdot v_2 + \int_{\Gamma_{out,2}} (C\xi_2 \mathbf{n}_2) \cdot v_2 + \int_{\Gamma_{int}} \{\{v_2\}\} \cdot [[C\xi_2]]. \end{cases}$$

Variational formulation

$$
\begin{cases}
\int_{\Omega_{h,1}} \rho \partial_t v_1 \cdot w_1 + \int_{\Omega_{h,2}} \rho \partial_t v_2 \cdot w_2 = -\int_{\Omega_{h,1}} \sigma_1 : \nabla w_1 - \int_{\Omega_{h,2}} \sigma_2 : \nabla w_2 \\
+\int_{\Gamma_{out,1}} (\sigma_1 \mathbf{n}_1) \cdot w_1 + \int_{\Gamma_{out,2}} (\sigma_2 \mathbf{n}_2) \cdot w_2 + \int_{\Gamma_{int}} \{\{\sigma_2\}\} : [[w_2]] \\
+\int_{\Gamma_{1/2}} [[\sigma w]], \\
\int_{\Omega_{h,1}} \partial_t \sigma_1 : \xi_1 + \int_{\Omega_{h,2}} \partial_t \sigma_2 : \xi_2 = \int_{\Omega_{h,1}} (C\xi_1) : \nabla v_1 - \int_{\Omega_{h,2}} (\boldsymbol{\nabla} \cdot (C\xi_2)) \cdot v_2 \\
+\int_{\Gamma_{out,2}} (C\xi_2 \mathbf{n}_2) \cdot v_2 + \int_{\Gamma_{int}} \{\{v_2\}\} \cdot [[C\xi_2]] \\
+\int_{\Gamma_{1/2}} [[(C\xi) v]].
\end{cases}
$$

Variational formulation

$$
\begin{cases}
\int_{\Omega_{h,1}} \rho \partial_t v_1 \cdot w_1 + \int_{\Omega_{h,2}} \rho \partial_t v_2 \cdot w_2 = -\int_{\Omega_{h,1}} \sigma_1 : \nabla w_1 - \int_{\Omega_{h,2}} \sigma_2 : \nabla w_2 \\
+ \int_{\Gamma_{out,1}} (\sigma_1 \mathbf{n}_1) \cdot w_1 + \int_{\Gamma_{out,2}} (\sigma_2 \mathbf{n}_2) \cdot w_2 + \int_{\Gamma_{int}} \{\{\sigma_2\}\} : [[w_2]] \\
+ \int_{\Gamma_{1/2}} \{\{\sigma\}\} : [[w]] + [[\sigma]] \cdot \{\{w\}\}, \\
\int_{\Omega_{h,1}} \partial_t \sigma_1 : \xi_1 + \int_{\Omega_{h,2}} \partial_t \sigma_2 : \xi_2 = \int_{\Omega_{h,1}} (C\xi_1) : \nabla v_1 - \int_{\Omega_{h,2}} (\boldsymbol{\nabla} \cdot (C\xi_2)) \cdot v_2 \\
+ \int_{\Gamma_{out,2}} (C\xi_2 \mathbf{n}_2) \cdot v_2 + \int_{\Gamma_{int}} \{\{v_2\}\} \cdot [[C\xi_2]] \\
+ \int_{\Gamma_{1/2}} \{\{v\}\} \cdot [[C\xi]] + \{\{C\xi\}\} : [[v]].
\end{cases}
$$

Variational formulation

$$\begin{cases}
\int_{\Omega_{h,1}} \rho \partial_t v_1 \cdot w_1 + \int_{\Omega_{h,2}} \rho \partial_t v_2 \cdot w_2 = -\int_{\Omega_{h,1}} \sigma_1 : \nabla w_1 - \int_{\Omega_{h,2}} \sigma_2 : \nabla w_2 \\
+ \int_{\Gamma_{out,1}} (\sigma_1 \mathbf{n}_1) \cdot w_1 + \int_{\Gamma_{out,2}} (\sigma_2 \mathbf{n}_2) \cdot w_2 + \int_{\Gamma_{int}} \{\{\sigma_2\}\} : [[w_2]] \\
+ \int_{\Gamma_{1/2}} \{\{\sigma\}\} : [[w]] \ \cancel{+ [[\sigma]] \cdot \{\{w\}\}}, \\
\int_{\Omega_{h,1}} \partial_t \sigma_1 : \xi_1 + \int_{\Omega_{h,2}} \partial_t \sigma_2 : \xi_2 = \int_{\Omega_{h,1}} (C\xi_1) : \nabla v_1 - \int_{\Omega_{h,2}} (\boldsymbol{\nabla} \cdot (C\xi_2)) \cdot v_2 \\
+ \int_{\Gamma_{out,2}} (C\xi_2 \mathbf{n}_2) \cdot v_2 + \int_{\Gamma_{int}} \{\{v_2\}\} \cdot [[C\xi_2]] \\
+ \int_{\Gamma_{1/2}} \{\{v\}\} \cdot [[C\xi]] \ \cancel{+ \{\{C\xi\}\} : [[v]]}.
\end{cases}$$

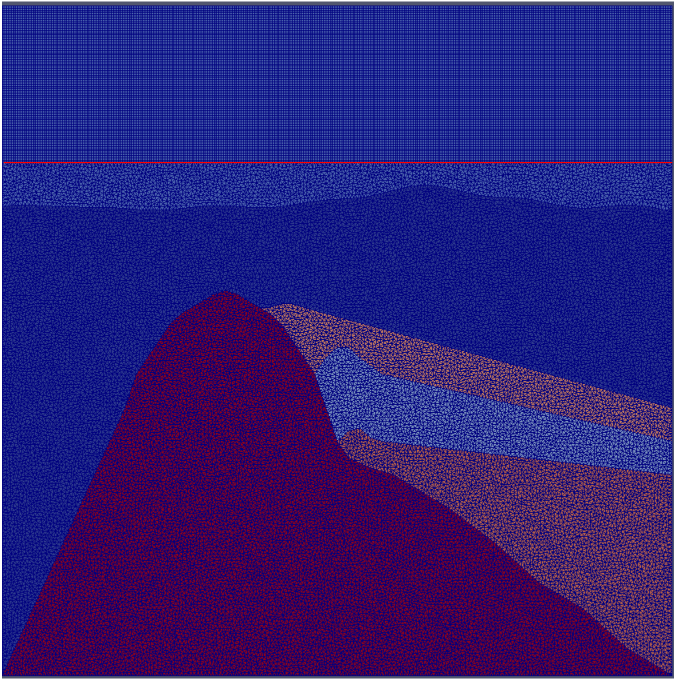
Energy study

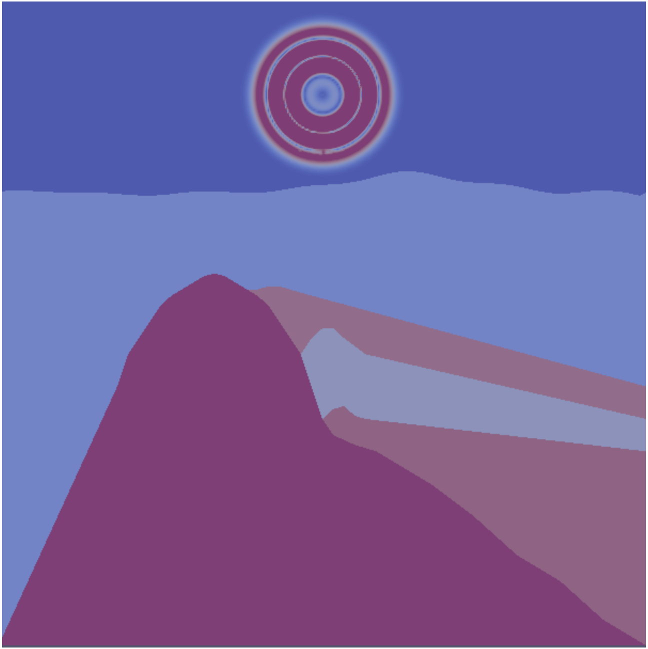
Goal : Show that our coupling preserves the energy

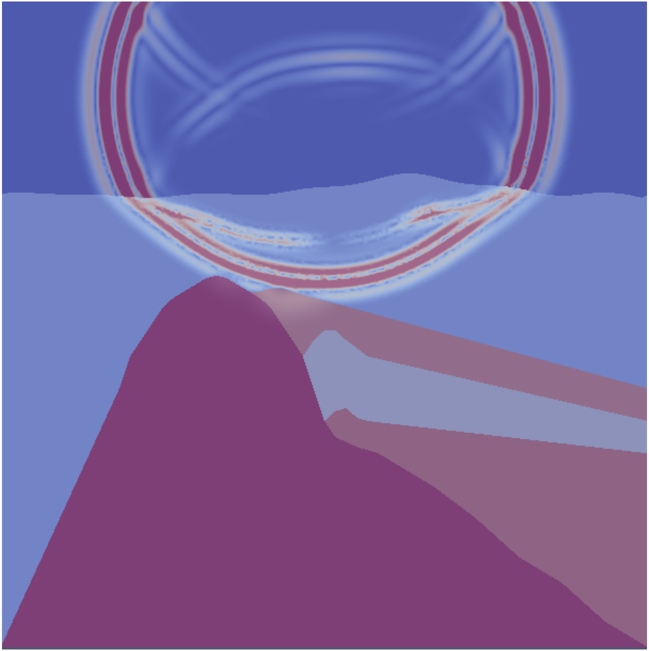
- We set $\xi_1 = \sigma_1, \xi_2 = \sigma_2, w_1 = v_1, w_2 = v_2$

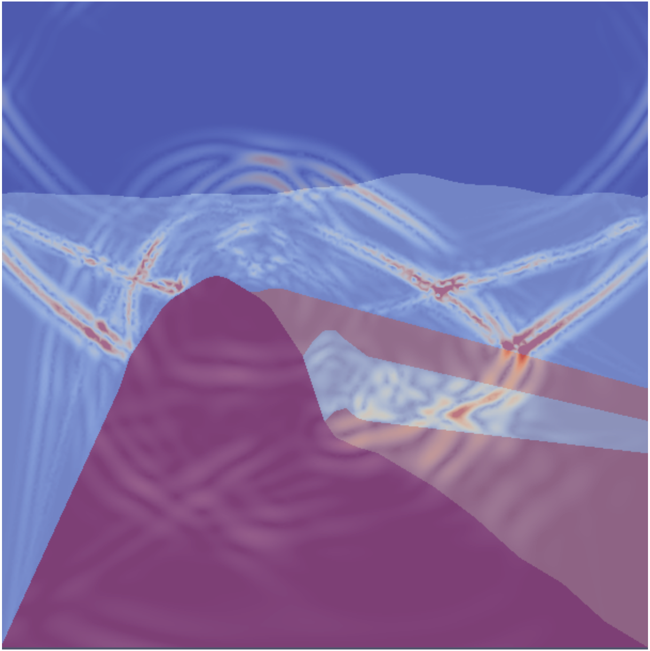
$$\frac{d}{dt}\mathbb{E} = 0$$

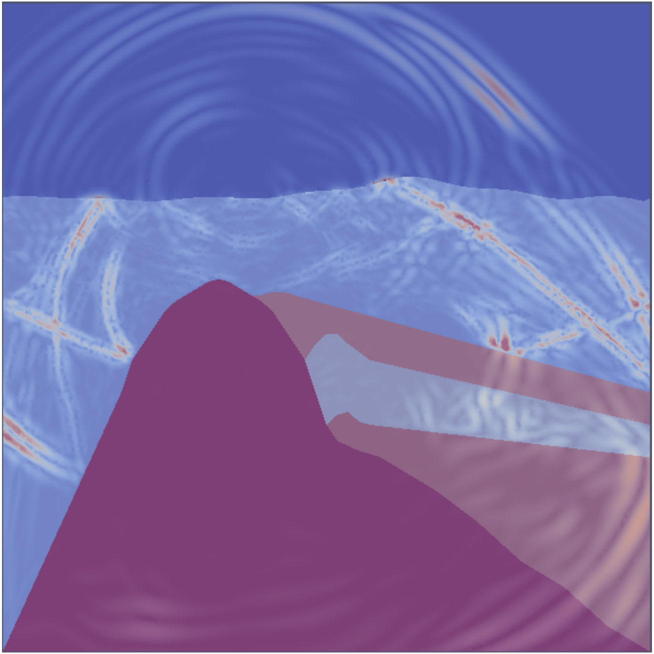
with $\mathbb{E} = \int_{\Omega_{h,1}} \rho \partial_t v_1 \cdot v_1 + \int_{\Omega_{h,2}} \rho \partial_t v_2 \cdot v_2 + \int_{\Omega_{h,1}} \partial_t \sigma_1 : \sigma_1 + \int_{\Omega_{h,2}} \partial_t \sigma_2 : \sigma_2$

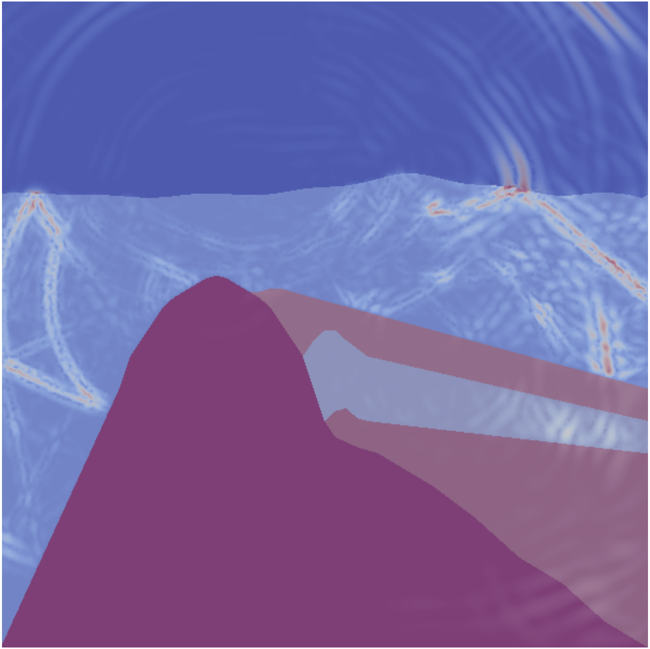












4 DGSEM vs DG

DGSEM vs DG

- Two dimensional Mesh with 74969 cells : 53969 triangles and 21000 quadrangles.

DGSEM vs DG

- Two dimensional Mesh with 74969 cells : 53969 triangles and 21000 quadrangles.
- Elastic isotropic case with order 2 on triangles (DG) and order 8 on the quadrangles (DG or SEM), point source at the center of the 3000x3000 domain.

	CFL	CPU-time	Nb of time steps
DG($\alpha = 0.5$)	1e-4	4148	10000
DG/SEM	1e-4	2645	10000

Figure: DG and DG/SEM comparison on an elastic isotropic case

DGSEM vs DG

- Two dimensional Mesh with 74969 cells : 53969 triangles and 21000 quadrangles.
- Elastic isotropic case with order 2 on triangles (DG) and order 8 on the quadrangles (DG or SEM), point source at the center of the 3000x3000 domain.
- "Salt dome" case with order 2 on the triangles (DG) and order 8 on the quadrangles (DG or SEM), point source on the top layer of water.

	CFL	CPU-time	Nb of time steps
DG($\alpha = 0.5$)	1e-4	4148	10000
DG/SEM	1e-4	2645	10000

Figure: DG and DG/SEM comparison on an elastic isotropic case

	CFL	CPU-time	Nb of time steps
DG($\alpha = 0.5$)	3e-5	38378	33333
DG/SEM	3e-5	22260	33333

Figure: DG and DG/SEM comparison on a "salt dome" case

5 3D extension

General settings

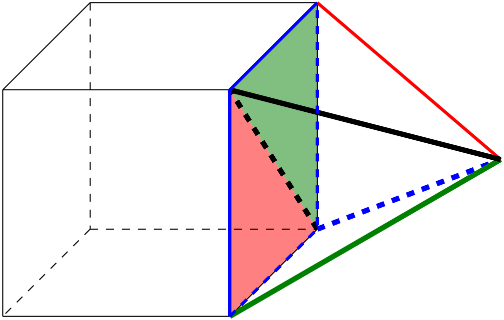


Figure: Hexa/Tet boundary configuration

- Only deal with a simple case of 3D hybrid meshes : one hexahedron has only two tetrahedra as neighbour.
- Extend SEM in 3D (basis functions...).
- Require introducing a new matrix which handles the rotation cases between two elements.

General settings

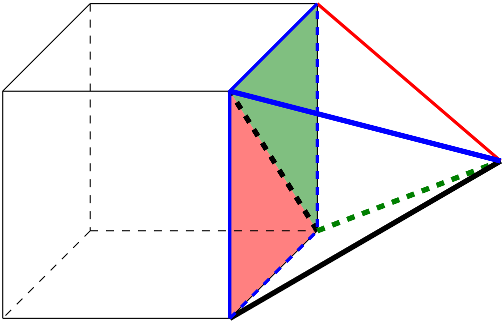


Figure: Hexa/Tet boundary configuration

- Only deal with a simple case of 3D hybrid meshes : one hexahedron has only two tetrahedra as neighbour.
- Extend SEM in 3D (basis functions...).
- Require introducing a new matrix which handles the rotation cases between two elements.

General settings

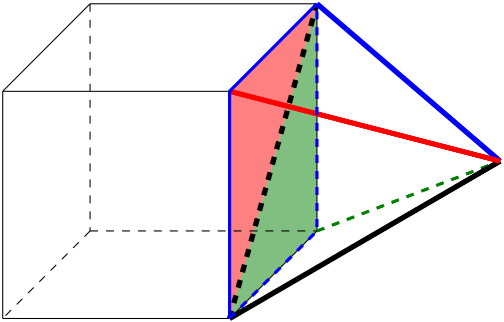


Figure: Hexa/Tet boundary configuration

- Only deal with a simple case of 3D hybrid meshes : one hexahedron has only two tetrahedra as neighbour.
- Extend SEM in 3D (basis functions...).
- Require introducing a new matrix which handles the rotation cases between two elements.

General settings

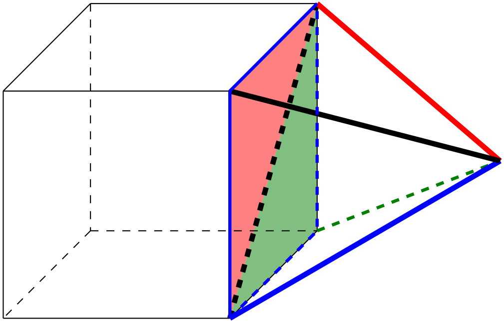
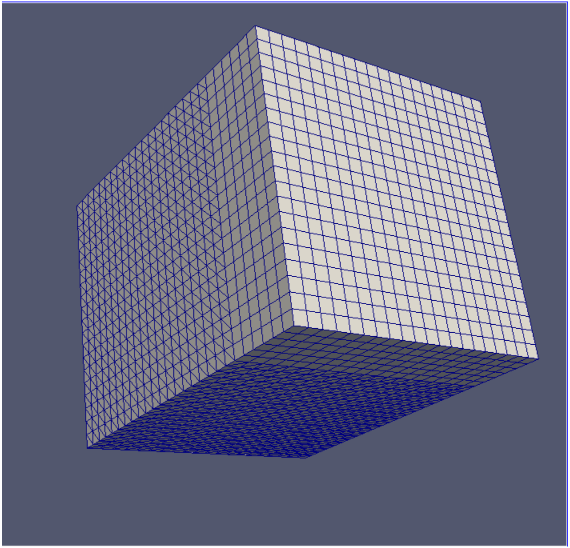
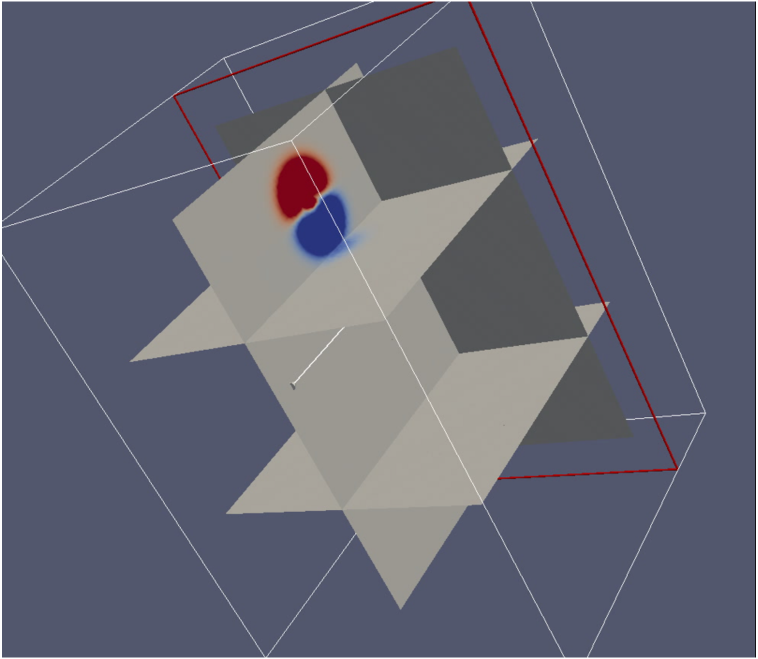
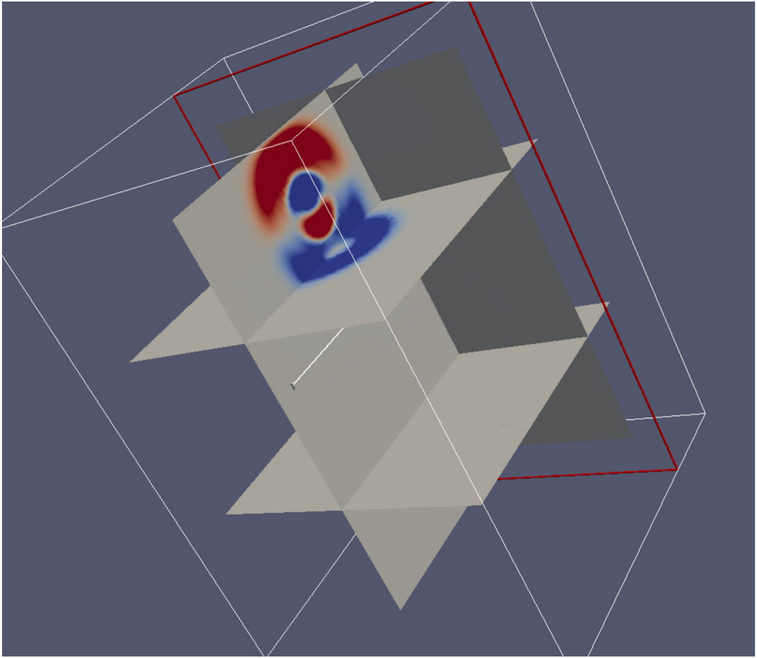


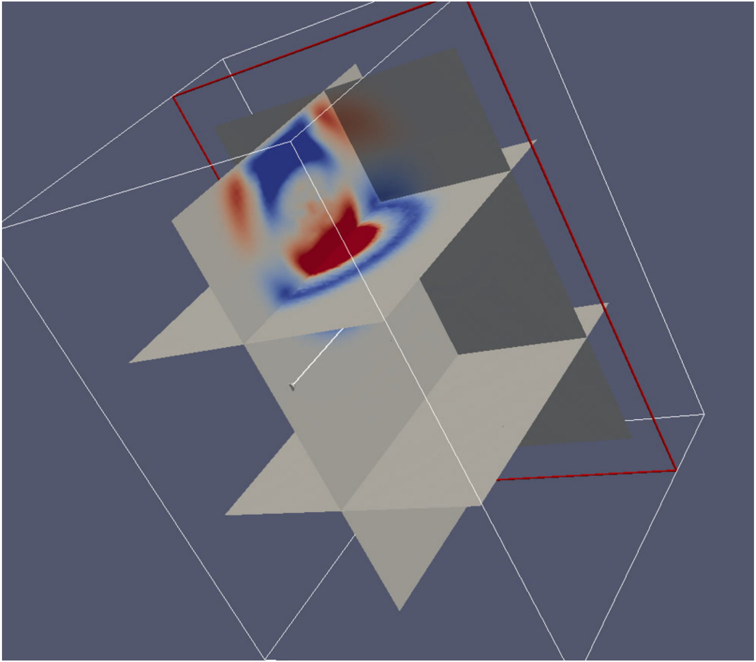
Figure: Hexa/Tet boundary configuration

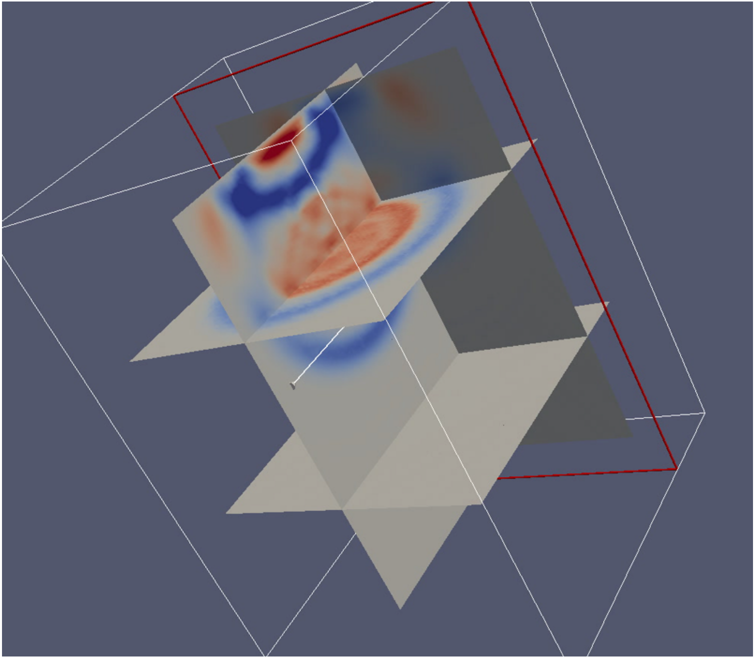
- Only deal with a simple case of 3D hybrid meshes : one hexahedron has only two tetrahedra as neighbour.
- Require introducing a new matrix which handles the rotation cases between two elements.

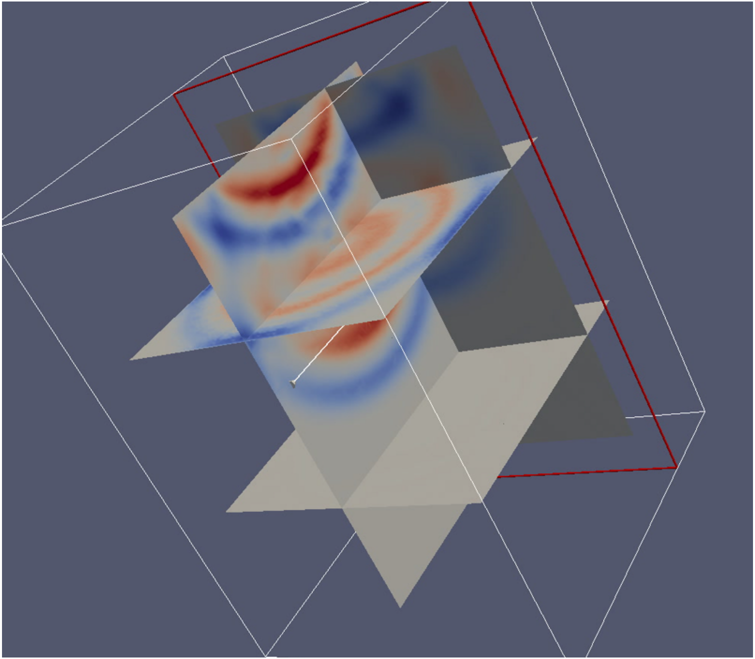


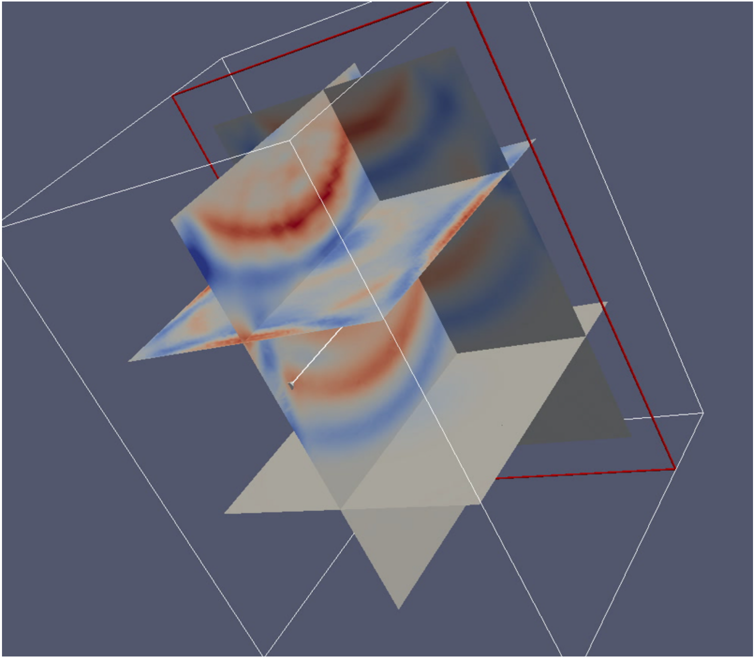


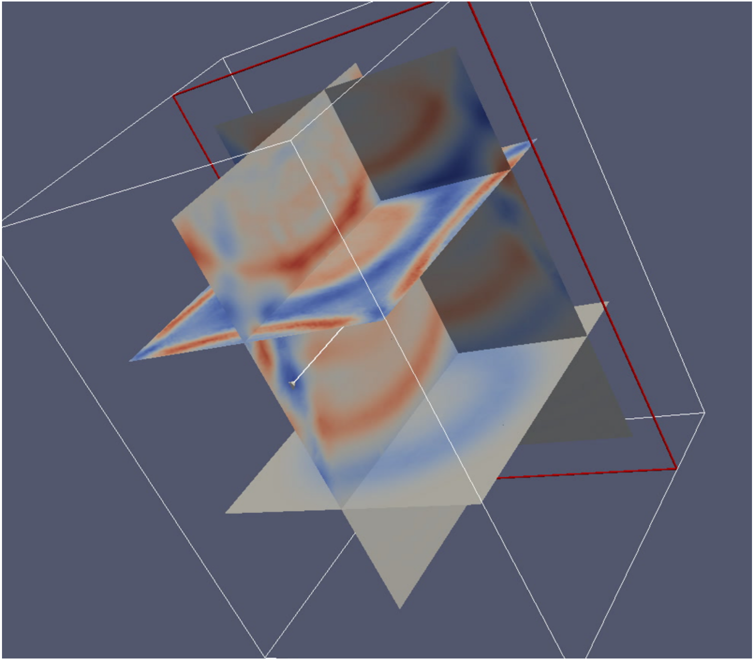






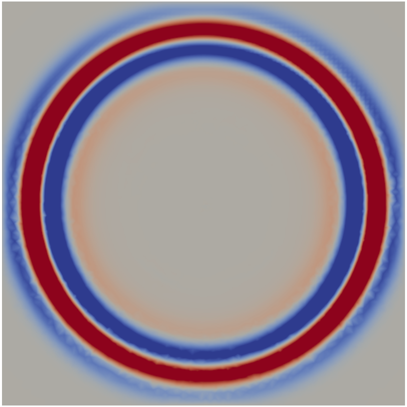




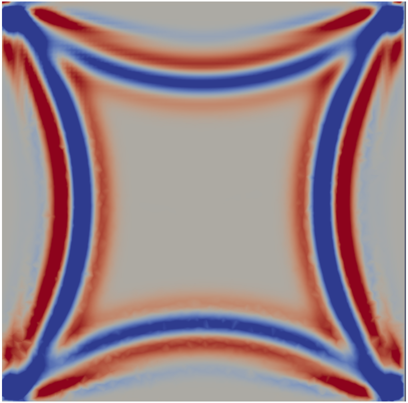


6. Perfectly Matched Layer(PML)
    - Mathematical formulation
    - ADE-PML (Auxiliary Differential Equation)
    - DGM vs DG/SEM simulation

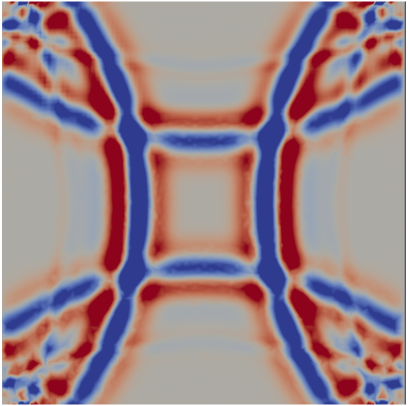
What is a PML and why using it?



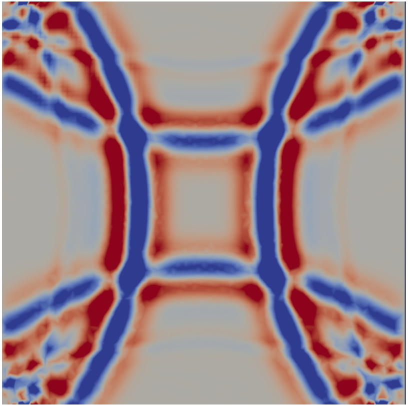
What is a PML and why using it?



What is a PML and why using it?

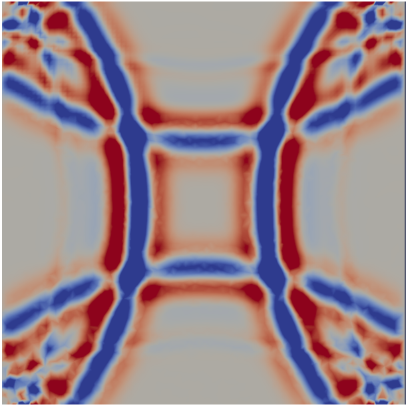


What is a PML and why using it?



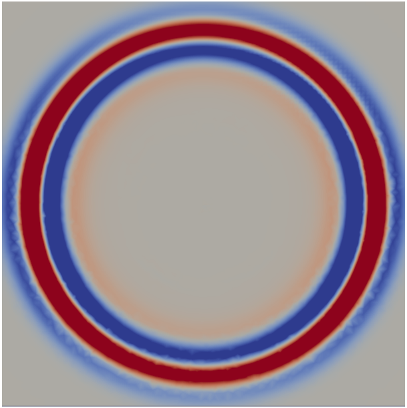
- Easy to implement

What is a PML and why using it?



- Easy to implement
- Not really adapted to the simulation of "infinite" domain

What is a PML and why using it?



What is a PML and why using it?

What is a PML and why using it?

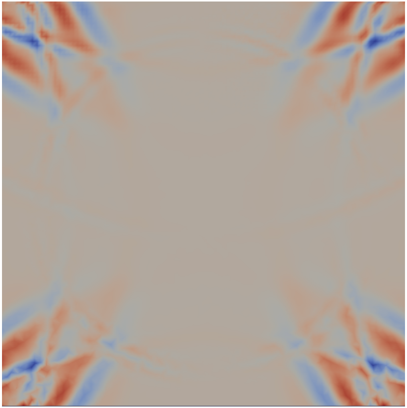
What is a PML and why using it?

- Possibility to simulate an infinite domain

What is a PML and why using it?

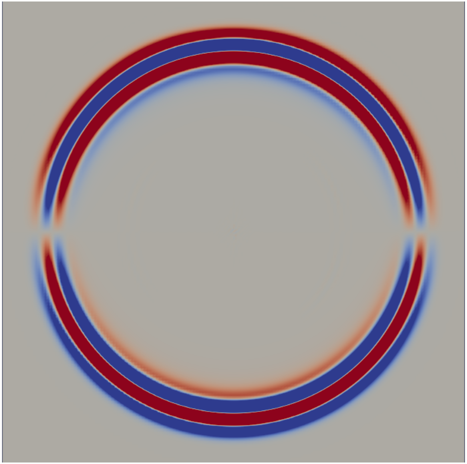
- Possibility to simulate an infinite domain
- Complicated to implement when the order of the condition increase

What is a PML and why using it?

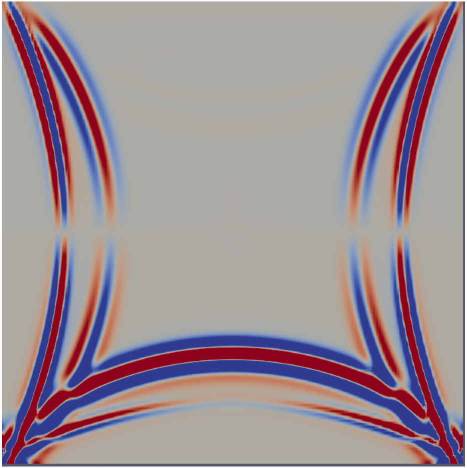


- Possibility to simulate an infinite domain
- Complicated to implement when the order of the condition increase
- Apparition of reflection when it comes to waves with grazing incidence

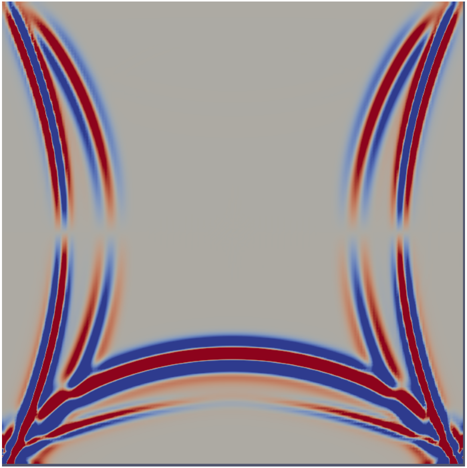
What is a PML and why using it?



What is a PML and why using it?

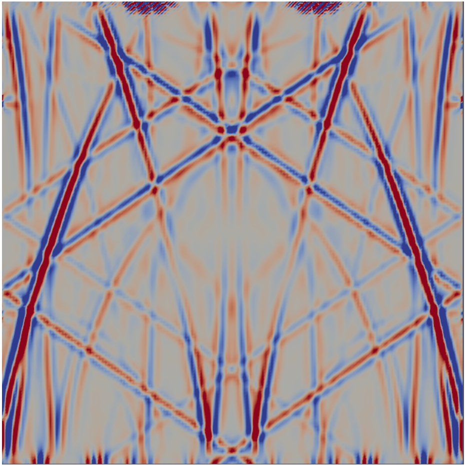


What is a PML and why using it?



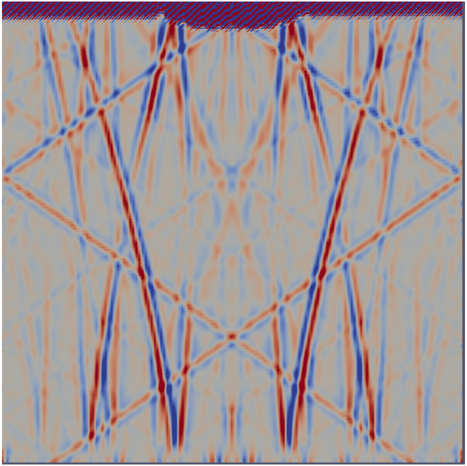
- Possibility to simulate an infinite domain

What is a PML and why using it?



- Possibility to simulate an infinite domain

What is a PML and why using it?



- Possibility to simulate an infinite domain
- Some stability problems on elastic case using DGM

Classic PML formulation

Step 1: Rewrite the system in the frequency domain

$$\begin{cases} i\omega\rho\mathbf{v} = \boldsymbol{\nabla}\cdot\underline{\underline{\sigma}} \\ \\ i\omega\underline{\underline{\sigma}} = \underline{\underline{C}} : \nabla\mathbf{v} \end{cases}$$

Classic PML formulation

Step 1: Rewrite the system in the frequency domain

$$\begin{cases} i\omega\rho\mathbf{v} = \boldsymbol{\nabla}\cdot\underline{\sigma} \\ \\ i\omega\underline{\sigma} = \underline{\underline{C}} : \nabla\mathbf{v} \end{cases}$$

Step 2: Introduce a new system of complex coordinates

Define d_z the damping :

$$\begin{cases} d_z > 0 \quad \text{if} \quad z \in \Omega_{PML} \\ d_z = 0 \quad \text{if} \quad z \notin \Omega_{PML} \end{cases}$$

Classic PML formulation

Step 2: Introduce a new system of complex coordinates

Define d_z the damping :

$$\begin{cases} d_z > 0 & \text{if} \quad z \in \Omega_{PML} \\ d_z = 0 & \text{if} \quad z \notin \Omega_{PML} \end{cases}$$

Step 2: Introduce a new system of complex coordinates

Define d_z the damping :

$$\begin{cases} d_z > 0 & \text{if} \quad z \in \Omega_{PML} \\ d_z = 0 & \text{if} \quad z \notin \Omega_{PML} \end{cases}$$

$$\tilde{z}(z) = z - \frac{i}{\omega} \int_0^z d_z(s) ds$$

Classic PML formulation

Step 2: Introduce a new system of complex coordinates

Define d_z the damping :

$$\begin{cases} d_z > 0 & \text{if} \quad z \in \Omega_{PML} \\ d_z = 0 & \text{if} \quad z \notin \Omega_{PML} \end{cases}$$

$$\tilde{z}(z) = z - \frac{i}{\omega} \int_0^z d_z(s) ds$$

Define the associate differential operator :

$$\partial_{\tilde{z}} = \frac{i\omega}{i\omega + d_z} \partial_z = 1 - \frac{d_z}{i\omega + d_z} \partial_z$$

Classic PML formulation

Step 3: Rewrite the system

$$
\begin{aligned}
i\omega\rho v_x &= \partial_x \sigma_{xx} + \partial_{\tilde{z}} \sigma_{xz}, \\
i\omega\rho v_z &= \partial_x \sigma_{xz} + \partial_{\tilde{z}} \sigma_{zz}, \\
i\omega\sigma_{xx} &= (\lambda + 2\mu)\partial_x v_x + \lambda \partial_{\tilde{z}} v_z, \\
i\omega\sigma_{zz} &= \lambda \partial_x v_x + (\lambda + 2\mu)\partial_{\tilde{z}} v_z, \\
i\omega\sigma_{xz} &= \mu(\partial_x v_z + \partial_{\tilde{z}} v_x).
\end{aligned}
$$

ADE-PML (Auxiliary Differential Equation)

Martin, Roland and Komatitsch, Dimitri and Gedney, Stephen D and Bruthiaux, Emilien and others
A high-order time and space formulation of the unsplit perfectly matched layer for the seismic wave equation using Auxiliary Differential Equations (ADE-PML)
Computer Modeling in Engineering and Sciences (CMES),2010

Introduction of $2^d + 1$ new variables $\psi_\star \in H^1 \rightarrow$ Simplify the implementation BUT add new equation to the linear system

ADE-PML (Auxiliary Differential Equation)

Martin, Roland and Komatitsch, Dimitri and Gedney, Stephen D and Bruthiaux, Emilien and others
A high-order time and space formulation of the unsplit perfectly matched layer for the seismic wave equation using Auxiliary Differential Equations (ADE-PML)
Computer Modeling in Engineering and Sciences (CMES),2010

Introduction of $2^d + 1$ new variables $\psi_\star \in H^1$ $\rightarrow$ Simplify the implementation BUT add new equation to the linear system

$$i\omega\rho v_z = \partial_x \sigma_{xz} + \partial_{\tilde{z}} \sigma_{zz}$$

ADE-PML (Auxiliary Differential Equation)

Martin, Roland and Komatitsch, Dimitri and Gedney, Stephen D and Bruthiaux, Emilien and others
A high-order time and space formulation of the unsplit perfectly matched layer for the seismic wave equation using Auxiliary Differential Equations (ADE-PML)
Computer Modeling in Engineering and Sciences (CMES),2010

Introduction of $2^d + 1$ new variables $\psi_\star \in H^1$ $\rightarrow$ Simplify the implementation BUT add new equation to the linear system

$$i\omega\rho v_z = \partial_x \sigma_{xz} + \partial_{\tilde{z}} \sigma_{zz}$$

$\downarrow$ Back to original coordinates

ADE-PML (Auxiliary Differential Equation)

Martin, Roland and Komatitsch, Dimitri and Gedney, Stephen D and Bruthiaux, Emilien and others
A high-order time and space formulation of the unsplit perfectly matched layer for the seismic wave equation using Auxiliary Differential Equations (ADE-PML)
Computer Modeling in Engineering and Sciences (CMES),2010

Introduction of $2^d + 1$ new variables $\psi_\star \in H^1$ $\rightarrow$ Simplify the implementation BUT add new equation to the linear system

$$i\omega\rho v_z = \partial_x \sigma_{xz} + \partial_{\tilde{z}} \sigma_{zz}$$

$\downarrow$ Back to original coordinates

$$i\omega\rho v_z = \partial_x \sigma_{xz} + \partial_z \sigma_{zz} - \frac{d_z}{d_z + i\omega} \partial_z \sigma_{zz}$$

ADE-PML (Auxiliary Differential Equation)

Martin, Roland and Komatitsch, Dimitri and Gedney, Stephen D and Bruthiaux, Emilien and others
A high-order time and space formulation of the unsplit perfectly matched layer for the seismic wave equation using Auxiliary Differential Equations (ADE-PML)
Computer Modeling in Engineering and Sciences (CMES),2010

Introduction of $2^d + 1$ new variables $\psi_\star \in H^1 \rightarrow$ Simplify the implementation BUT add new equation to the linear system

$$i\omega\rho v_z = \partial_x \sigma_{xz} + \partial_{\tilde{z}} \sigma_{zz}$$

$\downarrow$ Back to original coordinates

$$i\omega\rho v_z = \partial_x \sigma_{xz} + \partial_z \sigma_{zz} - \frac{d_z}{d_z + i\omega} \partial_z \sigma_{zz}$$

$\downarrow$ Back to time domain

Martin, Roland and Komatitsch, Dimitri and Gedney, Stephen D and Bruthiaux, Emilien and others
A high-order time and space formulation of the unsplit perfectly matched layer for the seismic wave equation using Auxiliary Differential Equations (ADE-PML)
Computer Modeling in Engineering and Sciences (CMES),2010

Introduction of $2^d + 1$ new variables $\psi_\star \in H^1$ $\rightarrow$ Simplify the implementation BUT add new equation to the linear system

$$i\omega\rho v_z = \partial_x \sigma_{xz} + \partial_{\tilde{z}} \sigma_{zz}$$

$\downarrow$ Back to original coordinates

$$i\omega\rho v_z = \partial_x \sigma_{xz} + \partial_z \sigma_{zz} - \frac{d_z}{d_z + i\omega} \partial_z \sigma_{zz}$$

$\downarrow$ Back to time domain

$$\rho \partial_t v_x = \partial_x \sigma_{xz} + \partial_z \sigma_{zz} - \frac{d_z}{d_z + \partial_t} \partial_z \sigma_{zz}$$

ADE-PML (Auxiliary Differential Equation)

Definition of the memory variable $\psi_{\sigma_{zz}}$:

$$\psi_{\sigma_{zz}} = -\frac{d_z}{d_z + \partial_t} \partial_z \sigma_{zz}$$

ADE-PML (Auxiliary Differential Equation)

Definition of the memory variable $\psi_{\sigma_{zz}}$:

$$\psi_{\sigma_{zz}} = -\frac{d_z}{d_z + \partial_t} \partial_z \sigma_{zz}$$

$$\rho \partial_t v_z = \partial_x \sigma_{xz} + \partial_z \sigma_{zz} + \psi_{\sigma_{zz}}$$

ADE-PML (Auxiliary Differential Equation)

Definition of the memory variable $\psi_{\sigma_{zz}}$:

$$\psi_{\sigma_{zz}} = -\frac{d_z}{d_z + \partial_t}\partial_z \sigma_{zz}$$

$$\rho \partial_t v_z = \partial_x \sigma_{xz} + \partial_z \sigma_{zz} + \psi_{\sigma_{zz}}$$

$$\partial_t \psi_{\sigma_{zz}} = -d_z \partial_z \sigma_{zz} - d_z \psi_{\sigma_{zz}}$$

ADE-PML (Auxiliary Differential Equation)

Definition of the memory variable $\psi_{\sigma_{zz}}$:

$$\psi_{\sigma_{zz}} = -\frac{d_z}{d_z + \partial_t} \partial_z \sigma_{zz}$$

$$\rho \partial_t v_z = \partial_x \sigma_{xz} + \partial_z \sigma_{zz} + \psi_{\sigma_{zz}}$$

$$\partial_t \psi_{\sigma_{zz}} = -d_z \partial_z \sigma_{zz} - d_z \psi_{\sigma_{zz}}$$

ADE with C-PML (Convolutional PML)

$$\partial_{\tilde{z}} = \left(1 - \frac{d_z}{\alpha_z + i\omega + d_z}\right) \partial_z$$

ADE-PML (Auxiliary Differential Equation)

Definition of the memory variable $\psi_{\sigma_{zz}}$:

$$\psi_{\sigma_{zz}} = -\frac{d_z}{d_z + \partial_t}\partial_z \sigma_{zz}$$

$$\rho \partial_t v_z = \partial_x \sigma_{xz} + \partial_z \sigma_{zz} + \psi_{\sigma_{zz}}$$

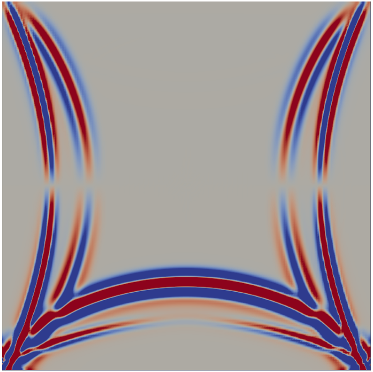
$$\partial_t \psi_{\sigma_{zz}} = -d_z \partial_z \sigma_{zz} - d_z \psi_{\sigma_{zz}}$$

ADE with C-PML (Convolutional PML)

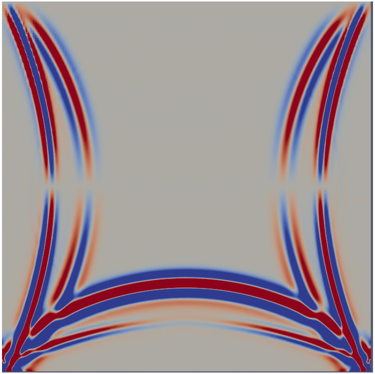
$$\partial_{\tilde{z}} = \left(1 - \frac{d_z}{\alpha_z + i\omega + d_z}\right)\partial_z$$

$$\partial_t \psi_{\sigma_{zz}} = -d_z \partial_z \sigma_{zz} - (d_z + \alpha_z)\psi_{\sigma_{zz}}$$

DG and DGSEM comparison

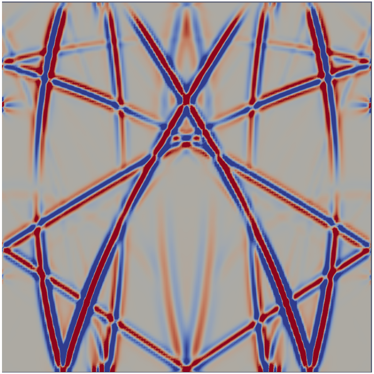


(a) t=1.5s

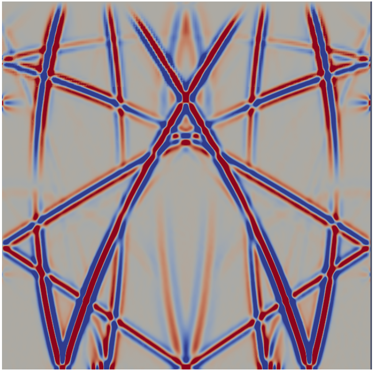


(b) t=1.5s

DG and DGSEM comparison

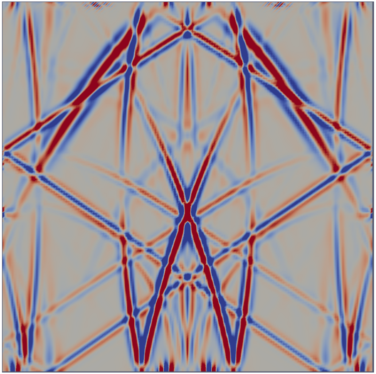


(c) t=3s

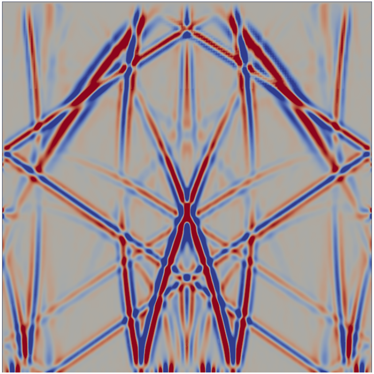


(d) t=3s

DG and DGSEM comparison

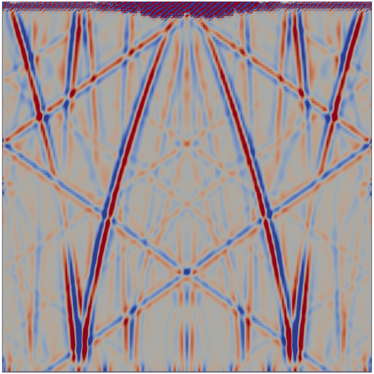


(e) t=5s

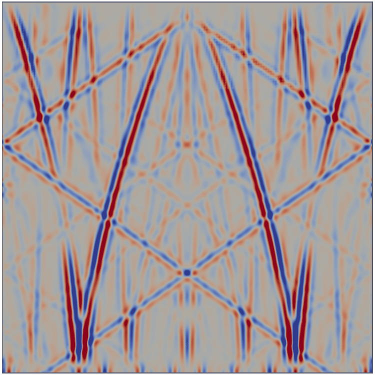


(f) t=5s

DG and DGSEM comparison

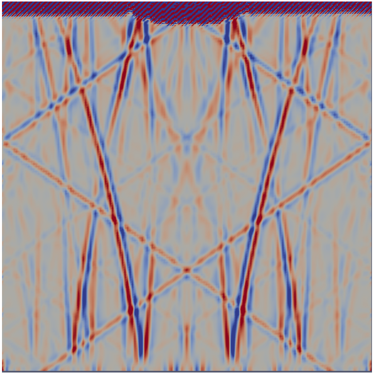


(g) t=8s

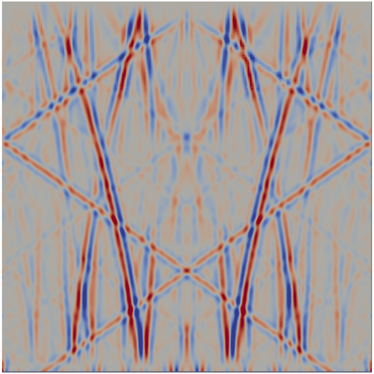


(h) t=8s

DG and DGSEM comparison



(i) t=10s



(j) t=10s

Conclusion and perspectives

Conclusion

1. SEM is more efficient on structured quadrangle mesh than DG
2. Build a variational formulation for DG/SEM coupling and find a CFL condition that ensures stability
3. Show the utility of using hybrid meshes and method coupling (reduce computational cost,...)

Achievements

- Implement DG/SEM coupling on the code (2D) ✓
- Develop DG/SEM coupling in 3D ✓
- Develop PML in the hexahedral part ✓

Thank you for your attention !

Questions?